Tutorial Week 5: SAT
Outline:

1. Encode Propositional Formulas in CNF with \textbf{Tsetin Transformation}

2. Model Bit-vector with SAT

3. Problem Solving with SAT: \textbf{Palindrome Sum}
Propositional Formulas

Propositional formulas can be defined as:

\[ f : \top | \nu | \neg f' | f_1 \land f_2 \quad (\bot \text{ can be expressed as } \neg \top) \]

Other operators can be defined as syntax sugars:

\[ f_1 \lor f_2 \quad := \neg (\neg f_1 \land \neg f_2) \]
\[ f_1 \Rightarrow f_2 \quad := \neg f_1 \lor f_2 \]
\[ f_1 \iff f_2 \quad := (f_1 \Rightarrow f_2) \land (f_2 \Rightarrow f_1) \]
\[ f_1 \oplus f_2 \quad := \neg (f_1 \iff f_2) \]
Conjunctive Normal Form (CNF)

• A formula $F$ in CNF is conjunction of clauses:

$$F := C_1 \land C_2 \ldots C_n$$

where a clause $C$ is a disjunction of literals:

$$C := l_1 \lor l_2 \ldots l_m$$

where a literal $l$ is either a variable or its complement:

$$l := v | \overline{v}$$
From Formula to CNF: Tsetin Transformation

- Sufficient to define transformation for:
  1. **Negation**:  
     \[ \neg a \rightarrow \bar{a} \]
  2. **Conjunction**:  
     \[ a \land b \rightarrow \text{a fresh literal } c \text{ and constraints} \]
     \[ (\bar{a} \lor \bar{b} \lor c) \land (\bar{c} \lor a) \land (\bar{c} \lor b) \]
Modeling Bit-vectors with SAT

Bit-vector (BV) is an array of 0 and 1s.

BV 1110 has the value of 14

We can model BV operations in CNF:

\[ \text{bv}_1 = \text{bv}_2 \quad \text{bv}_1 + \text{bv}_2 \]
Bit-vector Equality

\[ bv_1 = bv_2 \iff \]

\[ bv_1[0] \iff bv_2[0] \]
\[ bv_1[1] \iff bv_2[1] \]
\[ \ldots \]
\[ bv_1[k] \iff bv_2[k] \]
Bit-vector Addition (no overflow)

\[ \text{\(b v_1 + b v_2\)} \text{ returns a new bit-vector } b v_3 \text{ where (ripple-carry adder)} \]

\[
\begin{align*}
    b v_3[0] &= b v_1[0] \oplus b v_2[0] \\
    b v_3[1] &= b v_1[1] \oplus b v_2[1] \oplus \text{carry}_1 \\
    &\vdots \\
    b v_3[k] &= b v_1[k] \oplus b v_2[k] \oplus \text{carry}_k \\
    b v_3[k + 1] &= \text{carry}_{k+1}
\end{align*}
\]

\[\text{\(\text{carry}_{i+1} = \text{AtLeastTwo} (\text{carry}_i, b v_1[i], b v_2[i])\)}\]
Let’s prove the following statement with SAT:

\[ \text{bv}_1 > \text{bv}_3 \land \text{bv}_2 \geq \text{bv}_4 \]

\[ \Rightarrow \]

\[ \text{bv}_1 + \text{bv}_2 > \text{bv}_3 + \text{bv}_4 \]
Palindromic Sum

Given a natural number $n$, find two bit-vectors $bv_1$ and $bv_2$ such that

1. $bv_1$ and $bv_2$ are both palindromes:
   symmetrical, allow padding 0s to the left

2. The value of $bv_1 + bv_2$ is $n$
Solving Palindrome Sum with SAT

We already know how to encode bit-vector addition and equality!

We know the bit-vector representation of the input $n$!

We still need to model the conditions for palindrome:

**We need to consider different sizes of paddings for each bit-vector**

Finally, we need to call a SAT solver!